プロアクティブボット PBOT

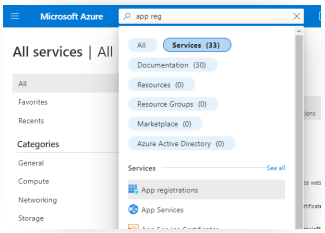
～ インストールと起動の方法 ～

目次

【作業 1】　App Administrations への登録 1
【作業 2】　Azure Bot の「登録」（作成） 2
【作業 3】　Azure AD SSO を利用するための設定 3
●1　API の公開（Expose an API） 4
●2　API のアクセス許可（API permissions） 6
●3　認証（Authentication） 7
●4　OAuth 接続設定（OAuth Connection Settings） 8
【作業 4】　MongoDB におけるコレクションの作成 9
【作業 5】　./.env および./demo/.env の設定 10
【作業 6】　./teamsAppManifest/manifest.json の設定 101
【作業 7】　デモボットの起動 112
【作業 8】　デモアプリの起動 123
【作業 9】　Teams アプリの組織へのアップロード 134
【作業 10】　ユーザーが Teams アプリをインストールする 145

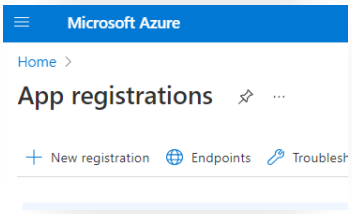
【作業 1】 App Administrations への登録

アプリ名を「PBOT1」(任意)、アカウントの種類を「Multiple Tenant」として登録する



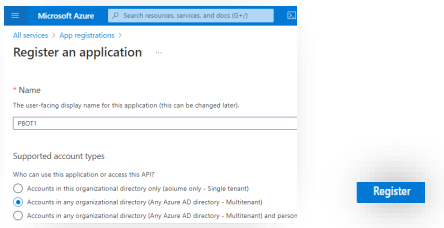
【ステップ 1】

Azure ポータルにログインして、検索欄に「app re」と入力し、[App registrations] を選択します。



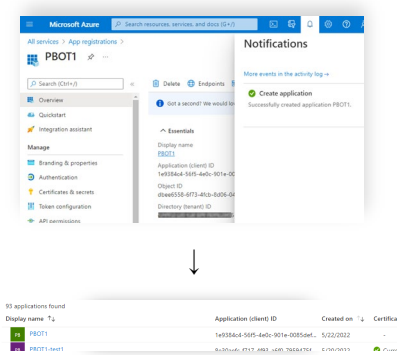
【ステップ 2】

App registrations 画面で、[New registration] をクリックします。



【ステップ 3】

Register an application 画面で、Name に「PBOT1」（任意値）と入力し、Supported account types の[Accounts in any organizational directory (Any Azure AD directory - Multitenant)]を選択してから、[Register]をクリックします。



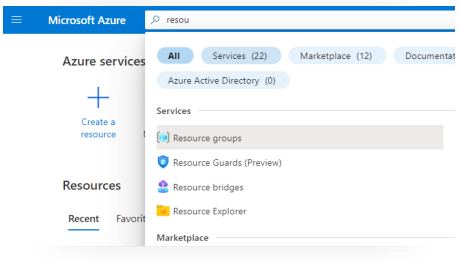
【ステップ 4】

Successfully created application PBOT1 という通知が表示されます。

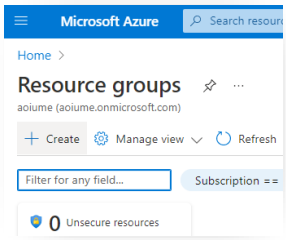
以上により、PBOT1 が App registrations に登録されました。

【作業 2】 Azure Bot の「登録」（作成）

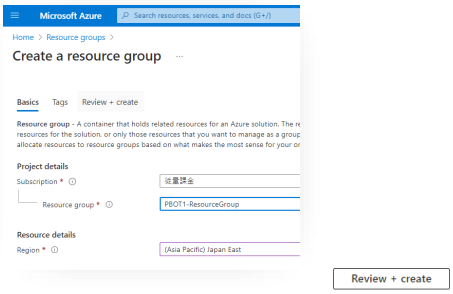
「PBOT1-ResourceGroup」（任意値）という名前の新しいリソースグループの配下に、
「PBOT1-AzureBot」（任意値）という名前の Azure Bot を作成する



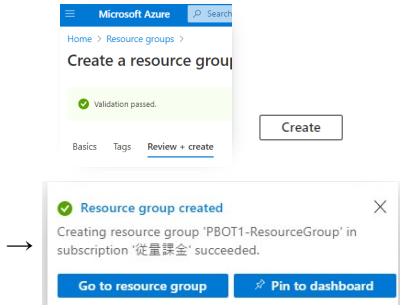
【ステップ 1】
Azure ポータルで、検索欄に「resou」と入力し、[Resource groups] を選択して、Resource groups 画面を開きます。



【ステップ 2】
[Create] をクリックして、Create a resource group 画面を開きます。

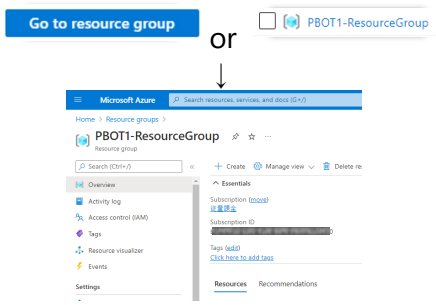


【ステップ 3】
Resource group に「PBOT1-ResourceGroup」（任意値）と入力し、Region プルダウンリストから「Japan East」（任意値）を選択してから、[Review+create]をクリックします。

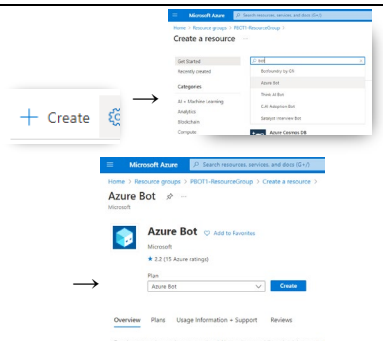


【ステップ 4】
validation passed と表示されるのを確認してから、[Create]をクリックします。Resource group created と表示されます。

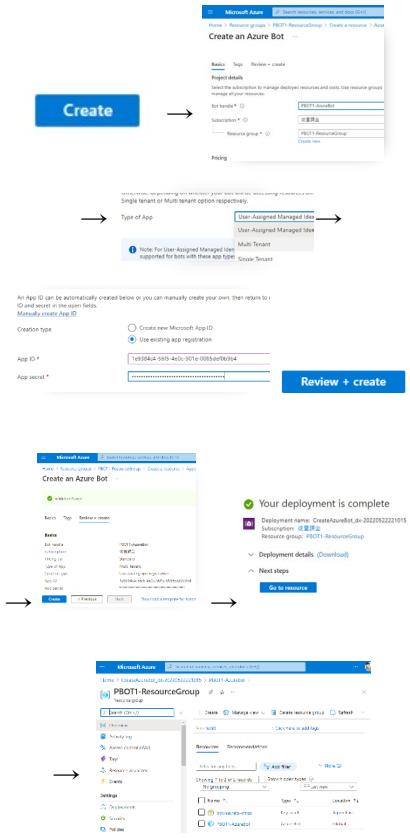
PBOT1-ResourceGroup リソースグループが作成されました。



【ステップ 5】
通知ダイアログの[Go to resource group]ボタンをクリックするか、または Resource groups 画面上のリソースグループ一覧から PBOT1-ResourceGroup を選択して、PBOT1-ResourceGroup 画面に移動します。



【ステップ 6】
[Create]ボタンをクリックして、Create a resource 画面に移動してから、検索欄に「bot」と入力し、[Azure Bot]を選択して Azure Bot 画面に移動します。

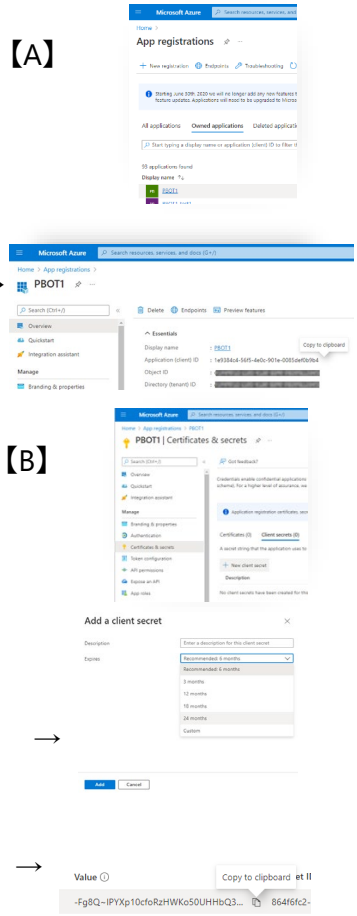


【ステップ 7】

Azure Bot 画面で、[Create]ボタンをクリックして、Create an Azure Bot 画面を開き、Bot Handle に「PBOT1-AzureBot」（任意値）と入力します。

つづいて、Type of App として[Multi Tenant]を選択すると、App ID 欄と App secret 欄が表示されます。右に示す、【補足ステップ A】、【補足ステップ B】の手順に沿って、これらの欄に値を入力し、[Review+create] ボタンをクリックします。Validation Passed という表示を確認して[Create]ボタンをクリックすると、デプロイメントが実行されます。

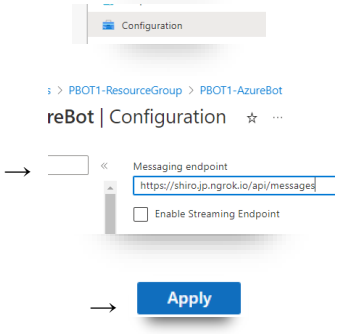
以上により、Azure Bot が作成されました。



【補足ステップ A】

別のウィンドウで App registrations 画面を開いて、アプリの一覧から PBOT1 を選択し、PBOT1 画面の **Overview** ブレードに移動します。このブレード上で、Application (client) ID をクリップボードにコピーして、【ステップ 7】の App ID 欄に貼り付けます。

【補足ステップ B】次に、PBOT1 画面の **Certificates & secrets** ブレードに移動してから、[New client secret]をクリックして、Add a client secret ダイアログの Expires から[24 months] （任意値）を選択してから[Add]ボタンをクリックします。作成された client secret をクリップボードにコピーして、【ステップ 7】の App secret 欄に貼り付けます。



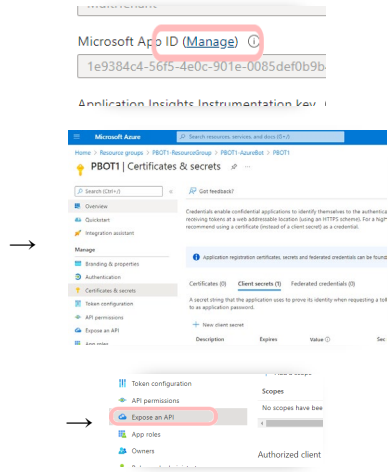
【ステップ 8】

Configuration ブレードの Messaging endpoint 欄に、「https://{{BOT_NGROK_DOMAINNAME}}/api/messages」（固定値）という形式の URI を入力してから、[Apply]ボタンを押します。

これにより、Bot Framework Adapter との送受信によりボットのハンドラを実現する、/api/messages エンドポイントが設定されました。

【作業 3】 Azure AD SSO を利用するための設定

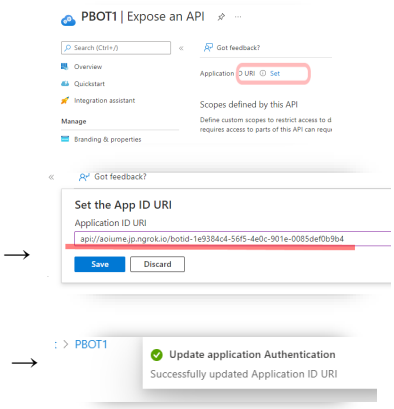
●1 API の公開（Expose an API）



【ステップ 1】

次に、**Configuration** ブレードで、Microsoft App ID の横に表示された、Manage というリンクをクリックします。【作業 1】で作成したアプリ PBOT1 の画面が表示されます。

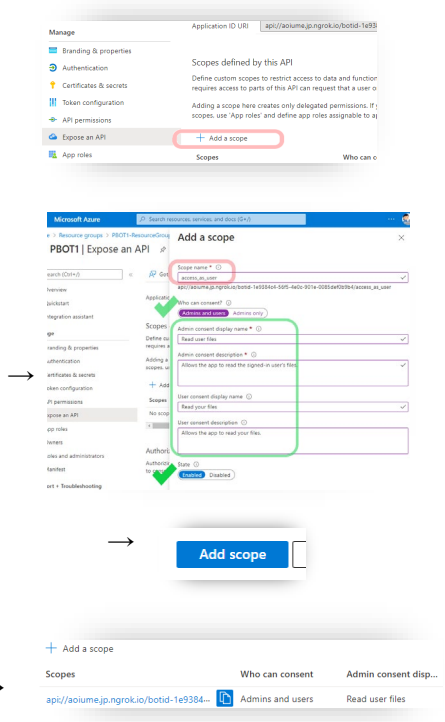
Expose an API をクリックして、**Expose an API** ブレードを開きます。



【ステップ 2】

Expose an API ブレードで、Application ID URI の横に表示された、Set というリンクをクリックします。プロンプトに対して、「api://{{TAB_NGROK_DOMAINNAME}}/botid-{{BOT_APPID}}」という形式の URI を入力してから、[Save]ボタンをクリックして URI を設定します。

タブとボットの両方を含む Teams アプリの場合は、この形式の URI が、AAD SSO におけるアプリの ID として使用されます。



【ステップ 3】

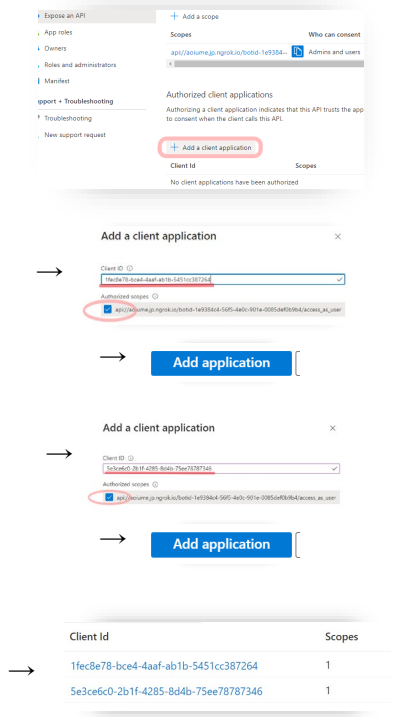
同じく **Expose an API** ブレードで、[+ Add a scope]ボタンをクリックして表示されるプロンプトに対して、まず、Scope name に「acccess_as_user」（固定値）と入力します。

次に、Who can consent、State についてそれぞれ、[Admin and users]、[Enabled]を選択します。

最後に、Admin consent display name から User consent description までについては、入力例のままの値を、それぞれ入力します。

[Add scope]ボタンをクリックすると、Scopes にスコープが追加されます。

以上により、Azure AD リソースへのアクセスに関して、「管理者とユーザーが認可を付与する」という動作を実現するためのエンドポイントと、認可を付与する際にプロンプトに表示されるテキストが、それぞれ設定されました。



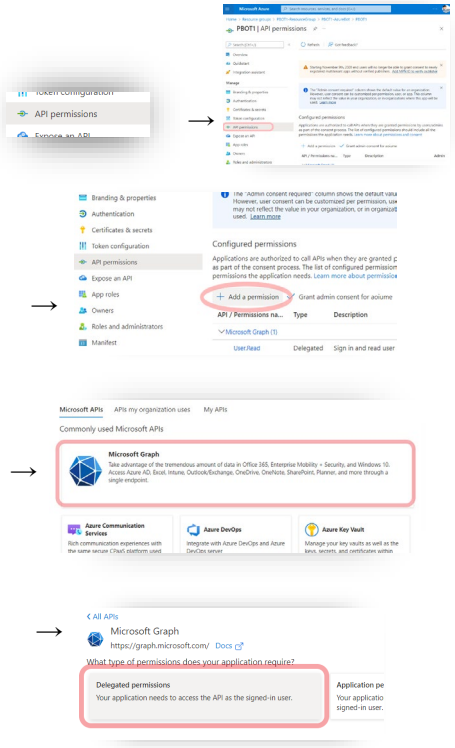
【ステップ 4】

同じく **Expose an API** ブレードで、[+ Add a client application]ボタンをクリックして表示されるプロンプトの Client ID に、デスクトップ版 Teams の AppID である「1fec8e78-bce4-4aaf-ab1b-5451cc387264」（固定値）を入力し、Authorized scopes のチェックボックスをオンにしてから、[Add application]ボタンをクリックします。

つづいて、ブラウザ版 Teams の AppID である「5e3ce6c0-2b1f-4285-8d4b-75ee78787346」（固定値）についても、同様に入力し、チェックボックスをオンにしてから、[Add application]ボタンをクリックします。

以上により、デスクトップ版とブラウザ版の Microsoft Teams が、スコープの範囲内で、Azure AD リソースへのアクセスが認可されたクライアントアプリケーションとして、設定されました。

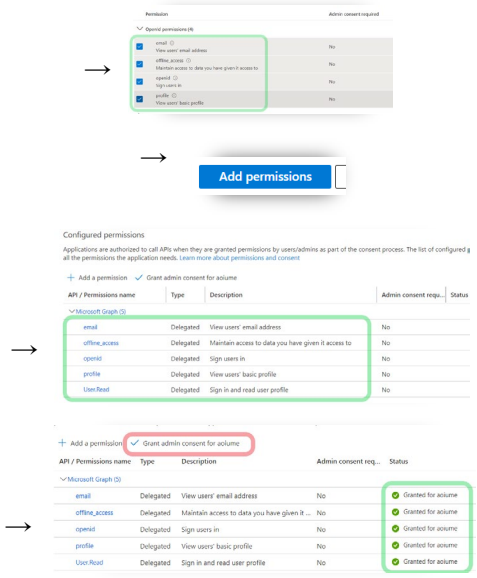
●2 API のアクセス許可（API permissions）



【ステップ 1】

[API permissions] を選択して、**API permissions** ブレードを開きます。つづいてこのブレードで、[+ Add a permission] ボタンをクリックすると、**Request API permissions** 画面に移動します。

Request API permissions 画面で、[Microsoft Graph]タイル、[Delegated permissions]タイルの順にクリックすると、アクセス許可（permissions）（「スコープ」とも呼ばれる）の一覧が表示されます。



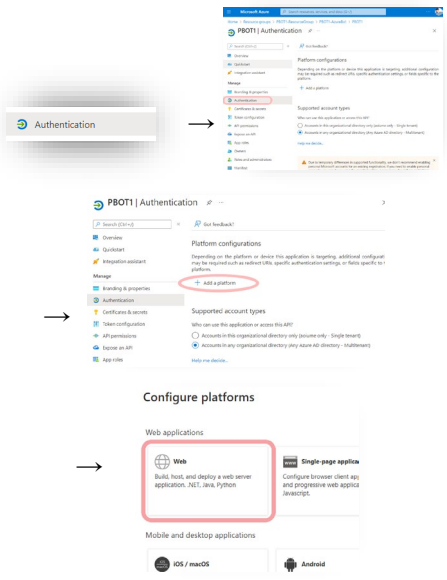
【ステップ 2】

アクセス許可の一覧のうち、email、offline_access、openid、profile という 4 つのチェックボックス（任意値）をオンにしてから、[Add permissions] ボタンをクリックすると、ブレード上の Configured permissions にこれらのアクセス許可が一覧表示されます。

この一覧に対して、[Grant admin consent for {{TENANT_NAME}}]ボタンをクリックすると、これらのアクセス許可のステータスがすべて、「✔ Grant for {{TENANT_NAME}}」に変わります。

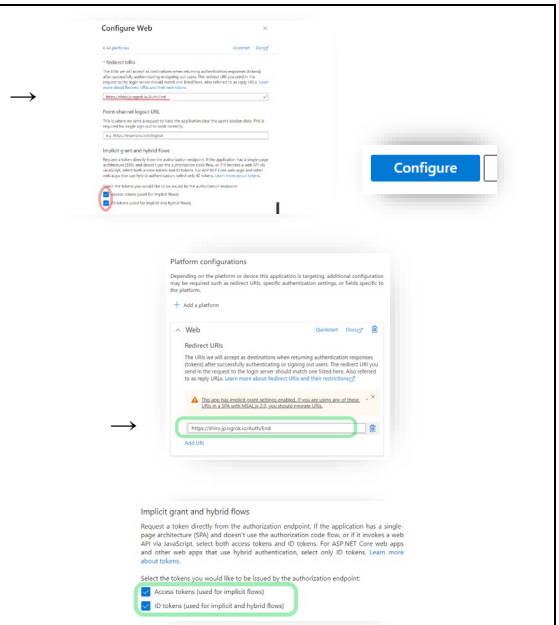
以上により、Microsoft Graph による Azure AD リソースへのアクセスのうち、email、offline_access、openid、profile といった範囲のアクセスについて、既に管理者から同意が得られており、あとはユーザーが「委任」について同意すれば認可（authorization）が得られる、という意味の「アクセス許可」が付与されました。

●3 認証（Authentication）



【ステップ 1】

[Authentication] を選択して、**Authentication** ブレードを開きます。このブレードで、[+ Add a platform] ボタンをクリックして表示される、Configure platforms というタイトルのプロンプトにおいて、[Web]タイルを選択すると、Configure Web というタイトルのプロンプトが表示されます。



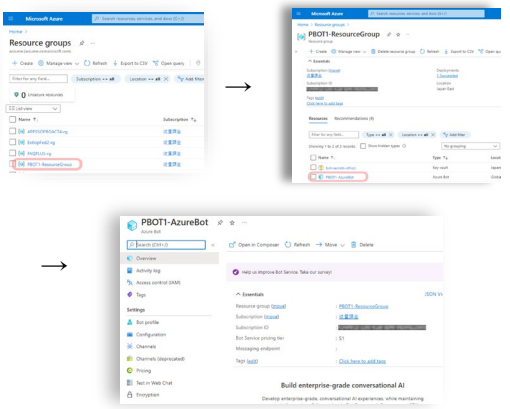
【ステップ 2】

このプロンプトにおいて、Redirect URIs に「https://{{BOT_NGROK_DOMAINNAME}}/Auth/End」（固定値）という形式の URI を入力します。

また、このプロンプト上の 2 つのチェックボックス、[Access tokens (used for implicit flows)]、および、[ID tokens (used for implicit and hybrid flows)] をオンにします。

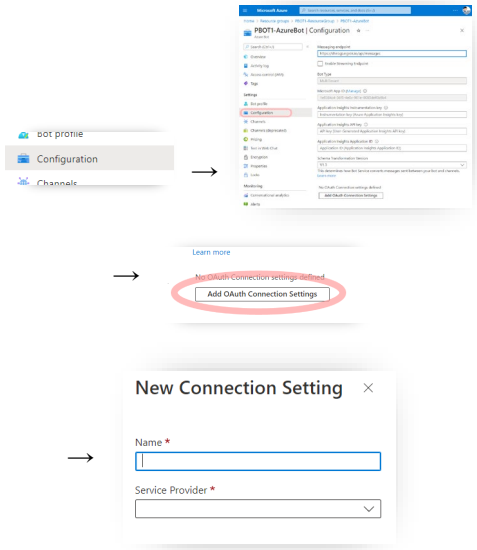
[Configure] ボタンをクリックすると、認証完了後のリダイレクト先のエンドポイントが設定されるとともに、認証プロセスにアクセストークンと ID トークンの両方が使用されるよう指定されます。

●4 OAuth 接続設定（OAuth Connection Settings）



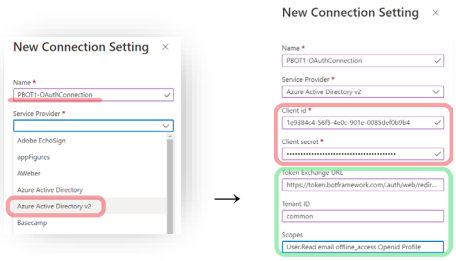
【ステップ 1】

Resource groups 画面のリソースグループの一覧から、PBOT1-ResourceGroup リソースグループの画面を開き、リソースの一覧から PBOT1-AzureBot の画面を開きます。



【ステップ 2】

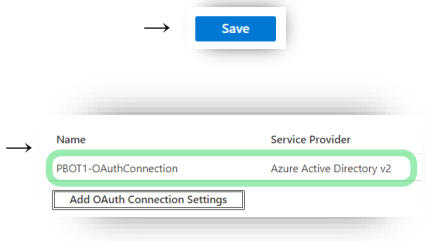
[Configuration] を選択して **Configuration** ブレードを開いてから、[Add OAuth Connection Settings] ボタンをクリックすると、New Connection Setting というタイトルのプロンプトが表示されます。



【ステップ 3】

Name に「PBOT1-OAuthConnection」（任意値）と入力し、Service Provider として [Azure Active Directory v2]（固定値）を選択すると、残りの欄が表示されます。このうち、Client id と Client secret には、【作業 1】で登録した PBOT1 の AppID と App Secret の値を入力します。

さらに、Token Exchange URL には Bot Framework が提供するエンドポイント「https://token.botframework.com/.auth/web/redirect」（固定値）、Tenant ID には「common」（マルチテナントの場合の固定値）、Scopes には【作業 3】●2 で設定したアクセス許可のスコープを空白文字で区切った値「User.Read email offline_access Openid Profile」を、それぞれ入力します。



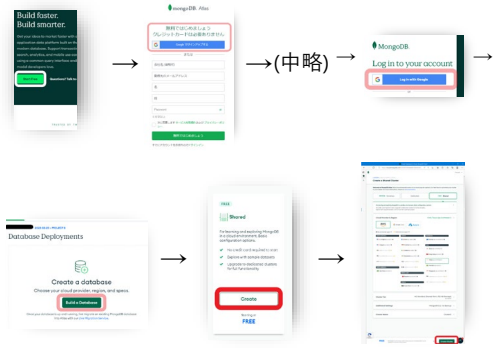
【ステップ 4】

[Save]ボタンをクリックすると、ブレード上の OAuth 接続設定の一覧に、Connection Setting プロンプト上で指定した名前 PBOT1-OAuthConnection が表示されます。

以上により、OAuth を使用して Azure AD に接続する際に指定する「OAuth 接続設定」が作成されました。デモボットにおいては、Graph クライアントを利用する際に、接続設定がこの名前（PBOT1-OAuthConnection）により特定されることになります（.env で指定する）。

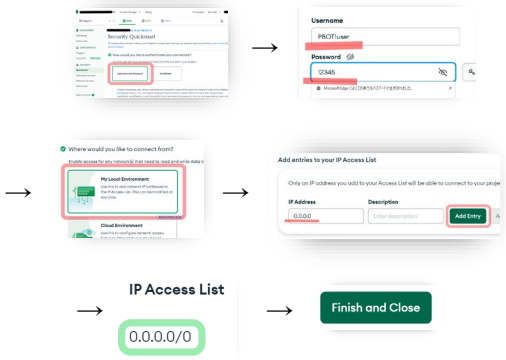
【作業 4】 MongoDB におけるコレクションの作成

MongoDB に、ユーザー名を「PBOT1user」（任意値）パスワードを「12345」（任意値）として、データベース「PBOT1DB」（任意値）およびその配下のコレクション「tusers」（固定値）を作成する。ならびに、デモボット、デモアプリでデータベースを利用するために、データベース接続文字列を取得する



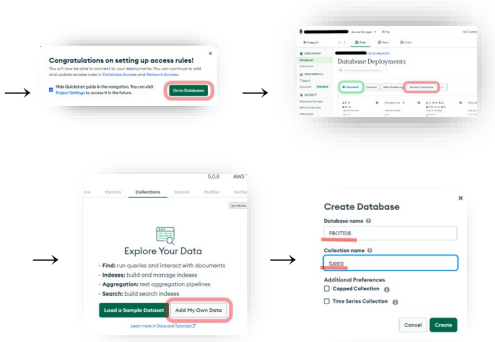
【ステップ 1】

MongoDB のページ（https://www.mongodb.com/）から、Atlas にサインアップし、それからサインインします。[Build a Database]ボタンをクリックすると表示される、3 つのオプションのうち Shared オプションの[Create]ボタンをクリックし、その次の画面でデフォルト値のままで[Create Cluster]ボタンをクリックします。



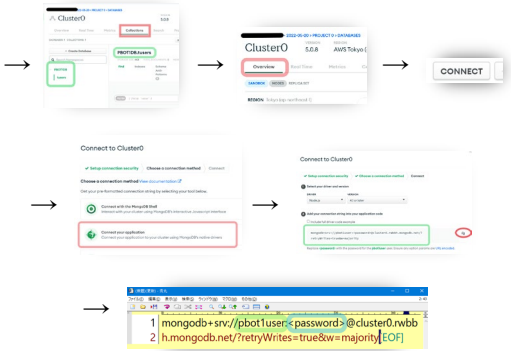
【ステップ 2】

Security Quickstart 画面で、[Username and Password]を選択して、Username に「PBOT1user」（任意値）、Password に「12345」（任意値）とそれぞれ入力します。また、[My Local Environment]を選択して、IP Address に「0.0.0.0」（任意の IP アドレスから接続する）と入力してから、[Add Entry]、[Finish and Close]の順にクリックします。



【ステップ 3】

[Go to Databases]をクリックすると、Cluster0 というクラスタがデプロイされたことが確認できます。[Browse Collections]タブを選択してから、[Add My Own Data]ボタンをクリックして表示される Create Database というプロンプトにおいて、Database name に「PBOT1DB」（任意値）、Collection name に「tusers」（固定値）とそれぞれ入力してから、[Create]ボタンをクリックします。



【ステップ 4】

Cluster0 クラスタに PBOT1DB.tusers というコレクションが作成されたことが確認できます。[Overview]タブに移動し、[CONNECT]ボタンをクリックしてから、[Connect your application]を選択して表示される画面から DB 接続文字列「mongodb+srv://pbot1user:<password>@cluster0.rwbbh.mongodb.net/?retryWrites=true&w=majority」をコピーして任意のテキストエディタに貼り付けておきます。

【作業 5】 ./.env および./demo/.env の設定

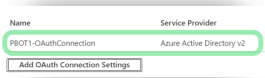
以下のプレースホルダの値を取得して置き換える

{{BOT_APPID}}	ボットの AppID
{{BOT_APP_PASSWORD}}	ボットの App Secret
{{AZUREBOT_OAUTH_CONNECTION_NAME}}	OAuth 接続設定名
{{MONGODB_CONNECTION_STRING}}	DB 接続文字列
{{TAB_NGROK_DOMAINNAME}}	タブ用の NGROK ドメイン名

【補足ステップ A】
別のウィンドウで App registrations 画面を開いて、アプリの一覧から PBOT1 を選択し、PBOT1 の画面の **Overview** ブレードに移動します。このブレード上で、Application (client) ID をクリップボードにコピーして、【ステップ 7】の App ID 欄に貼り付けます。

【補足ステップ B】
App registrations 画面の **Certificates & secrets** ブレードに移動してから、[New client secret]をクリックして、Add a client secret ダイアログの Expires から[24 months] （任意値）を選択してから[Add]ボタンをクリックします。作成された client secret をクリップボードにコピーして、【ステップ 7】の App secret 欄に貼り付けます。

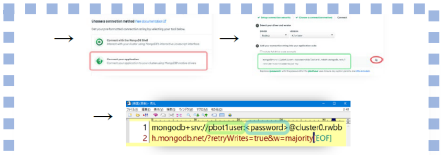
【ステップ 1】
{{BOT_APPID}}と{{BOT_APP_PASSWORD}}については、【作業 2】の【補足ステップ A】【補足ステップ B】と同じ方法(要領)で取得します。



【ステップ 4】
[Save]ボタンをクリックすると、ブレード上の OAuth 接続設定の一覧に、Connection Setting プロンプト上で指定した名前 PBOT1-OAuthConnection が表示されます。

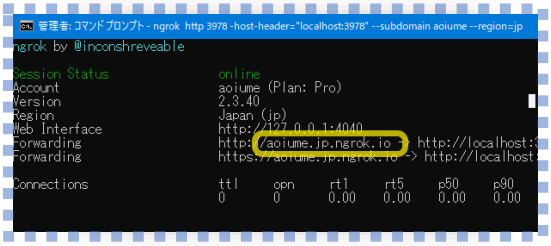
以上により、OAuth を使用して Azure AD に接続する際に指定する「OAuth 接続設定」が作成されました。デモボットにおいては、Graph クライアントを利用する際に、接続設定がこの名前（PBOT1-OAuthConnection）により特定されることになります（.env で指定する）。

【ステップ 2】
{{AZUREBOT_OAUTH_CONNECTION_NAME}}については、【作業 3】の●4【ステップ 4】に基づき、「PBOT1-OAuthConnection」が値となります。



【ステップ 4】
Cluster0 クラスタに PBOT1DB.tusers というコレクションが作成されたことが確認できます。
[Overview]タブに移動し、[CONNECT]ボタンをクリックしてから、[Connect your application]を選択して表示される画面から DB 接続文字列「mongodb+srv://pbot1user:<password>@cluster0.rwbbh.mongodb.net/?retryWrites=true&w=majority」をコピーして任意のテキストエディタに貼り付けておきます。

【ステップ 3】
{{MONGODB_CONNECTION_STRING}}については、【作業 4】の【ステップ 4】に基づき、「mongodb+srv://pbot1user:12345@cluster0.rwbbh.mongodb.net/?retryWrites=true&w=majority」が値となります。

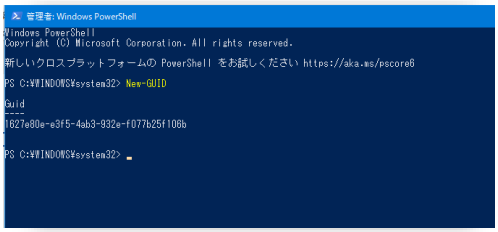


【ステップ 4】
{{TAB_NGROK_DOMAINNAME}}については、NGROK から購入して取得したドメイン名のうち、タブ用(デモアプリ用)に使用しているものを設定します。たとえば、上に示した例の場合は、「aoiume.jp.ngrok.io」が値となります。

【作業 6】 ./teamsAppManifest/manifest.json の設定

以下のプレースホルダの値を取得して置き換える

{{TAB_APPID}}	タブの AppID
{{BOT_APPID}}	ボットの AppID
{{TAB_NGROK_DOMAINNAME}}	タブ用の NGROK ドメイン名



【ステップ 1】

{{TAB_APPID}} は、任意の GUID を作成して使用します。PowerShell で New-GUID を実行するなどして取得します。

別のウィンドウで App registrations 画面を開いて、アプリの一覧から PBOT1 を選択し、PBOT1 の画面の **Overview** ブレードに移動します。このブレード上で、Application (client) ID をクリップボードにコピーして、【ステップ 7】の App ID 欄に貼り付けます。

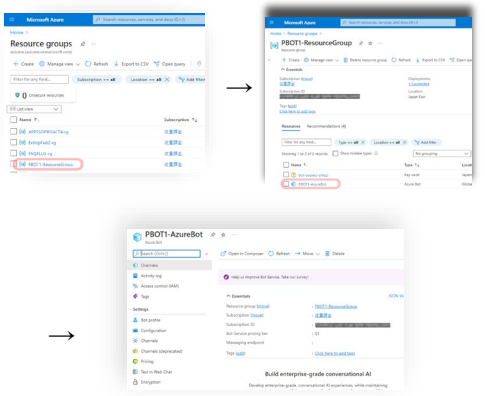
{{TAB_NGROK_DOMAINNAME}}については、NGROK から購入して取得したドメインネームのうち、タブ用(デモアプリ用)に使用しているものを設定します。たとえば、上に示した例の場合は、「aoiume.jp.ngrok.io」が値となります。

【ステップ 2】

{{BOT_APPID}}と{{TAB_NGROK_DOMAIN NAME}}については、【作業 5】の【ステップ 1】と【ステップ 4】をそれぞれ参照してください。

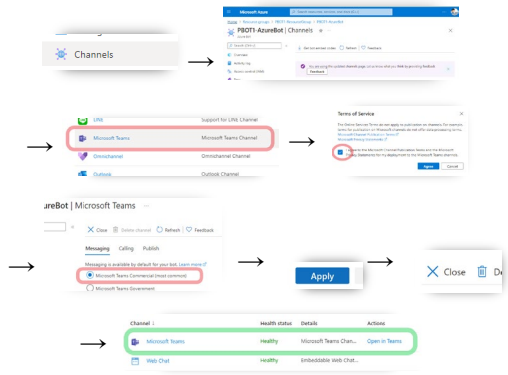
【作業 7】 デモボットの起動

最初に行うインストール作業として：(a) Teams へのチャンネル接続、(b) npm install
随時行う起動作業として：(c) ngrok の起動、(d) npm run dev



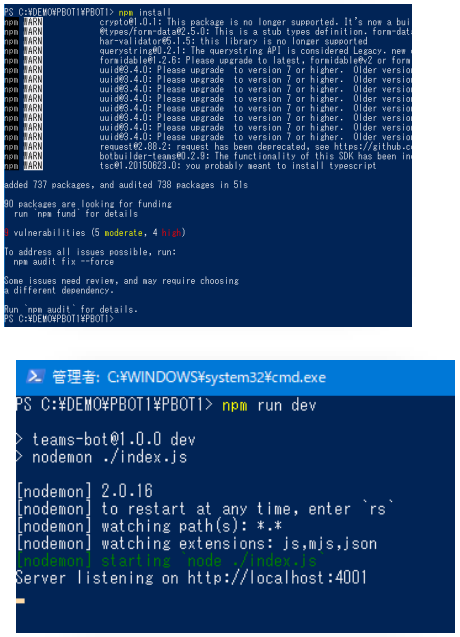
【ステップ 1】

Resource groups 画面のリソースグループの一覧から、**PBOT1-ResourceGroup** リソースグループの画面を開き、リソースの一覧から **PBOT1-AzureBot** の画面を開きます。



【ステップ 2】

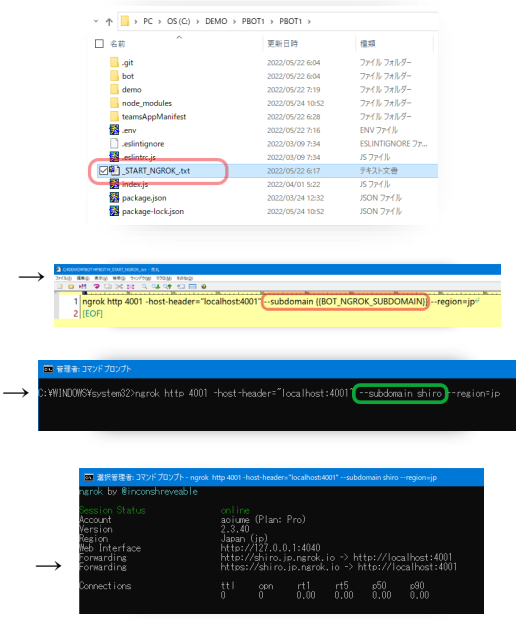
[Channels]をクリックして **Channels** ブレードを開きます。Available Channels の一覧にある **[Microsoft Teams]** をクリックしてから、Terms of Service に対して**[Agree]**の意思表示をします。**[Microsoft Teams Commercial (most common)]** をオンにしてから、**[Apply]**、**[× Close]**の順にクリックすると、Channels に Microsoft Teams が追加されます。



【ステップ 3】

PowerShell 上でデモのトップフォルダに移動して、「**npm install**」を実行します。NodeJS コードのモジュールがインストールされます。

つづいて、「**npm run dev**」を実行すると、デモボットが起動します。

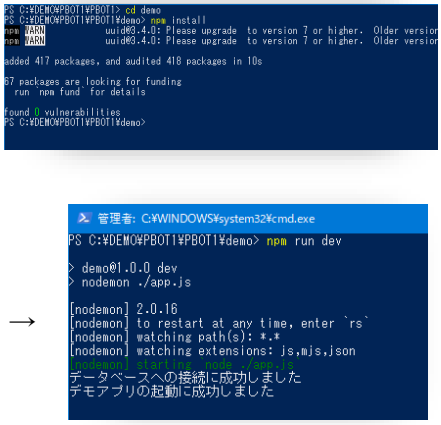


【ステップ 4】

デモのトップフォルダにある**_START_NGROK_.txt** ファイルをテキストエディタで開き、プレースホルダー**{{BOT_NGROK_SUBDOMAIN}}**をボット用のサブドメインに置き換えてできた文字列を、コマンドプロンプトからデモのトップフォルダ上で実行します。

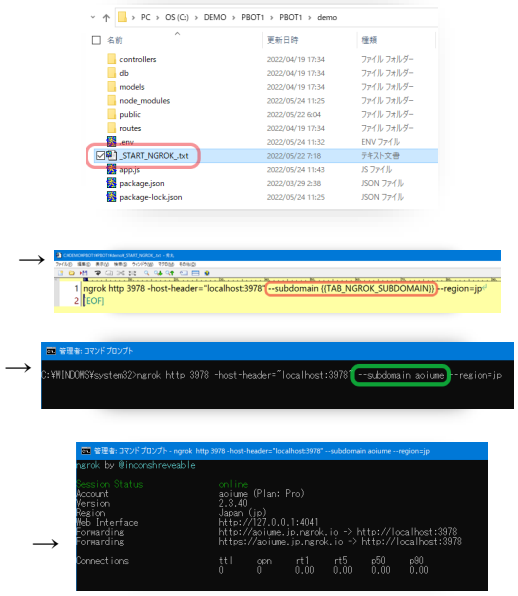
【作業 8】 デモアプリの起動

最初に行うインストール作業として：(a) npm install
随時行う起動作業として：(b) ngrok の起動、(c) npm run dev



【ステップ 1】
PowerShell 上でデモアプリのフォルダに移動して、「**npm install**」を実行します。NodeJS コードのモジュールがインストールされます。

つづいて、「**npm run dev**」を実行すると、デモアプリが起動します。

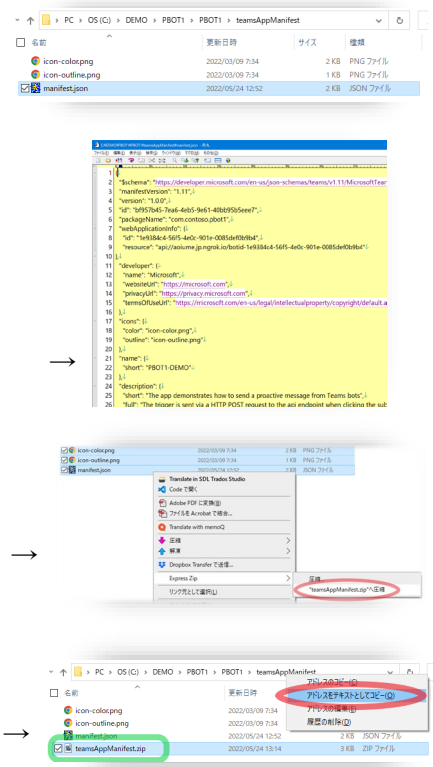


【ステップ 2】
デモアプリのフォルダにある_START_NGROK_.txt ファイルをテキストエディタで開き、プレースホルダー {{TAB_NGROK_SUBDOMAIN}}をタブ用のサブドメインに置き換えてできた文字列を、コマンドプロンプトからデモアプリのフォルダ上で実行します。

【ステップ 3】
ブラウザのアドレス欄にデモアプリの URL 「**https://{{TAB_NGROK_DOMAINNAME}}**」を入力して **Enter** キーを押すと、デモアプリが表示されます。

【作業 9】 Teams アプリの組織へのアップロード

組織のアプリカタログに Teams アプリをアップロードする



【ステップ 1】

manifest.json の**{{プレースホルダー}}**がすべて値に置き換わっていることを確認し、**2 つのアイコンファイル**とあわせて圧縮した**パッケージ ZIP ファイル**を作成します。**パッケージ ZIP ファイル**の存在場所がわかるように、フォルダのアドレスをテキストとしてコピーしておきます。

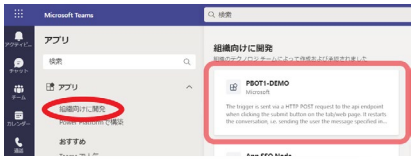
【ステップ 2】

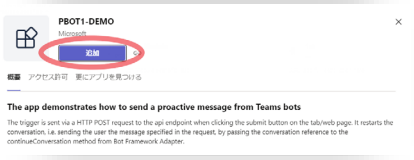
Teams 上で、**[アプリ]**、**[アプリを管理]**、**[アプリをアップロード]**の順にクリックすると、**[組織のアプリカタログにアプリをアップロードします]**タイルが表示されます。このタイルをクリックして表示されるダイアログに対して、【ステップ 1】でクリップボードにコピーしておいたフォルダのアドレスを貼り付けて、**パッケージ ZIP ファイル**を選択し、**[開く]**ボタンをクリックします。

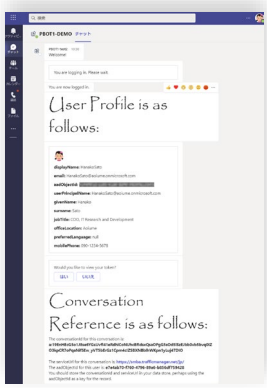
「組織に向けて開発」画面のアプリ一覧に、デモボットが追加されました。

【作業 10】ユーザーが Teams アプリをインストールする

Teams のチャット画面にデモボットをインストールする







【ステップ 1】

ユーザーが Teams 上で、**[アプリ]**、**[組織向けに開発]**の順にクリックしてから、アプリのタイルをクリックすると、アプリをインストールするためのプロンプトが表示されます。プロンプト上の **[追加]** ボタンをクリックします。

Teams のチャット画面にデモボットがインストールされました。